Claims

I claim:

Sub A11

1. The structure of a subpipelined translation embodiment providing binary compatibility between a base architecture and migrant architecture of a VLIW architecture comprising:

a VLIW architecture comprising a base architecture and a migrant architecture and having a base execution mode and a migrant execution mode;

a fetch packet retrieved from memory, the fetch packet having an operating mode in dependence upon the execution mode at the time the request was made to the memory for the fetch packet;

a shared datapath by both the base and migrant architectures for parsing said base architecture mode and migrant architecture mode fetch packets into execute packets and for dispatching those base execute packets to the appropriate base architecture decode of the execute hardware;

a migrant architecture control circuit for dispatching execute packet instructions having a migrant execution mode to a migrant architecture decode;

execute hardware for executing execute packet instructions on execution units and having a base architecture decode and a migrant architecture decode for decoding said base architecture instructions and said migrant architecture instructions, respectively, in dependence upon the execution mode of the fetch packet of the instructions being decoded, prior to executing;

a multiplexer having at least two inputs and one machine word output wherein one input is the output of said migrant architecture decode and the other input is the output of said base architecture decode, said multiplexer choosing in dependence upon the operating mode of said fetch packet;

machine words for controlling the execution hardware units.

2. The structure according to Claim 1, and further comprising a third input to said multiplexer wherein said third input is a no operation instruction

3. The structure according to Claim 1, wherein said machine word also controls registers.

4. The structure according to Claim 1, wherein said machine word controls a global register file, which supplies operands to all hardware execution units and accepts results of all hardware execution units.

5. The structure according to Claim 4, wherein said machine word controls local register files that supply operands to either local execution hardware functional units or neighbor hardware execution functional units subsequent to said machine word controlling said global register file.

6. The structure according Claim 5, wherein said machine word controls the various types of execution hardware that evaluate functions on the operands to produce the results of said hardware execution units subsequent to said machine word controlling said local register files.

7. The structure according to Claim 1, wherein the base and migrant architecture decode units translates opcodes to the control signals required to execute the specified instructions on the execution hardware functional units.

8. The structure according to claim 1, and further comprising said migrant architecture control circuit for issuing no-operation instruction to preserve the semantics of the instruction in the migrant architecture.

9. The structure according to Claim 1, wherein said VLIW architecture is a Digital signal Processor (DSP).

Cont All

10. A method of providing binary compatibility between a base architecture and migrant architecture of a VLIW architecture comprising the steps of:

executing a base execution mode and a migrant execution mode on a base architecture and a migrant architecture, respectively, on a VLIW architecture;

providing a fetch packet retrieved from a memory, the fetch packet having an operating mode in dependence upon the execution mode at the time the request was made to the memory for the fetch packet;

parsing said base architecture mode and migrant architecture mode fetch packets into execute packets and dispatching those base execute packets to the appropriate base architecture decode of the execute hardware on a shared datapath by both the base and migrant architectures;

dispatching execute packet instructions having a migrant execution mode to a migrant architecture decode on a migrant architecture control circuit;

executing execute packet instructions on execution units of execute hardware said execute hardware having a base architecture decode and a migrant architecture decode for decoding said base architecture instructions and said migrant architecture instructions, respectively, in dependence upon the execution mode of the fetch packet of the instructions being decoded, prior to executing;

choosing, in dependence upon the operating mode of said fetch packet, between the output of said migrant architecture decode and the output of said base architecture decode in a multiplexer having one machine word output;

controlling the execution hardware units with said machine word.

11. The method according to Claim 10, and further comprising choosing between the output of said migrant architecture decode and the output of said base architecture decode input and a no operation instruction

Cont All

12. The method according to Claim 10, and further comprising controlling registers with said machine word.

13. The method according to Claim 10, and further comprising controlling a global register file with said machine word, which supplies operands to all hardware execution units and accepts results of all hardware execution units.

14. The structure according to Claim 13, and further comprising controlling local register files that supply operands to either local execution hardware functional units or neighbor hardware execution functional units subsequent to said controlling said global register file.

15. The method according Claim 14, and further comprising controlling the various types of execution hardware that evaluate functions on the operands to produce the results of said hardware execution units subsequent to controlling said local register files.

16. The method according to Claim 10, and further comprising translating opcodes to the control signals required to execute the specified instructions on the execution hardware functional units within the base and migrant architecture decode units.

17. The method according to claim 10, wherein said VLIW architecture is a Digital Signal Processor (DSP).

18. The method according to Claim 10 and further comprising the step of issuing no-operation instruction from said migrant architecture control circuit, to preserve the semantics of the instructions in the migrant architecture.